2. Growth of Function

- Typically, problems become computationally intensive as the input size grows.
- We look at input sizes large enough to make only the order of the growth of the running time relevant for the analysis and comparison of algorithms.
- Hence we are studying the asymptotic efficiency of algorithms.
- So far our analysis showed that:
  - Merge-Sort has a running time of $\Theta(n \lg n)$
  - Insertion-Sort has a running time of $\Theta(n^2)$
- We like to make this notion more precise.

\[ \Theta - \text{Notation} \]

- Definition: Let $g(n)$ be an asymptotically non-negative function on the natural numbers.
  \[
  \Theta(g(n)) = \{ f(n) \mid \exists c_1 > 0, c_2 > 0, n_0 \in \text{Nat} \cdot \forall n \geq n_0 : 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \}
  \]
- Function $f(n)$ belongs to $\Theta(g(n))$ if it can be sandwiched between $c_1 g(n)$ and $c_2 g(n)$ for some constants $c_1$, $c_2$, for all $n$ greater than some $n_0$.
- In this case, we say that $g(n)$ is an asymptotically tight bound for $f(n)$.
- We write $f(n) = \Theta(g(n))$ for $f(n) \in \Theta(g(n))$
Examples for $\Theta$ ...

• $n^2 / 2 - 3 n = \Theta(n^2)$
  We have to determine $c_1 > 0$, $c_2 > 0$, $n_0 \in \text{Nat}$ such that:
  
  $c_1 n^2 \leq n^2 / 2 - 3 n \leq c_2 n^2$ for any $n \geq n_0$

  Dividing by $n^2$ yields:
  
  $c_1 \leq 1 / 2 - 3 / n \leq c_2$

  This is satisfied for $c_1 = 1 / 14$, $c_2 = 1 / 2$, $n_0 = 7$.

• $6 n^3 \neq \Theta(n^2)$
  We would have to determine $c_1 > 0$, $c_2 > 0$, $n_0 \in \text{Nat}$ such that:
  
  $c_1 n^2 \leq 6 n^3 \leq c_2 n^2$ for any $n \geq n_0$

  which cannot exist.

... Examples for $\Theta$

• $a n^2 + b n + c = \Theta(n^2)$ provided $a > 0$
  For example, take $c_1 = a / 4$, $c_2 = 7 a / 4,$
  $n_0 = 2 \max (|b|/a, \sqrt{(|c|/a)}).

• In general, if $a_m > 0$ then
  
  $\sum_{i=0}^{m} a_i n^i = \Theta(n^m)$
Properties of $\Theta$

- Assume $f(n)$ and $g(n)$ are asymptotically positive:
  
  - $f(n) = \Theta(g(n)) \land g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$ (Transitivity)
  
  - $f(n) = \Theta(f(n))$ (Reflexivity)
  
  - $f(n) = \Theta(g(n)) \iff g(n) = \Theta(f(n))$ (Symmetry)
  
  - $\max(f(n), g(n)) = \Theta(f(n) + g(n))$ (Maximum)

If $f(n)$ and $g(n)$ are the running times of the two branches of an `if`-statement, then this can be used to get a tight bound on the worst case running time of the whole statement if nothing is known about the condition, e.g. depends on unknown input.

---

$O$-Notation

- **Definition**: Let $g(n)$ be an asymptotically non-negative function on the natural numbers.
  
  $$O(g(n)) = \{ f(n) \mid \exists c > 0, n_0 \in \text{Nat} \cdot \forall n \geq n_0 \cdot 0 \leq f(n) \leq c \cdot g(n) \}$$

- In this case, we say that $g(n)$ is an **asymptotic upper bound** for $f(n)$.

- $\Theta$ is stronger than $O$:
  
  $f(n) = \Theta(g(n)) \Rightarrow f(n) = O(g(n))$, or $\Theta(g(n)) \subseteq O(g(n))$

- We write $f(n) = O(g(n))$ for $f(n) \in O(g(n))$
Examples for $O$

- $a n^2 + b n + c = O(n^2)$ provided $a > 0$
  since it is also $\Theta(n^2)$.

- $a n + b = O(n^2)$ provided $a > 0$

- $n \log n + n = O(n^2)$

- $\lg^k n = O(n)$ for all $k \in \Nat$

- $O$ can be used for an upper bound of the running time for worst-case input (and hence for any input).

- Note: Some books use $O$ to informally describe tight bounds. Here we use $\Theta$ for tight bounds and $O$ for upper bounds.

Ω-Notation

- **Definition**: Let $g(n)$ be an asymptotically non-negative function on the natural numbers.
  $$\Omega(g(n)) = \{f(n) \mid \exists c > 0, n_0 \in \Nat \cdot \forall n \geq n_0 \cdot 0 \leq c g(n) \leq f(n)\}$$

- In this case, we say that $g(n)$ is an asymptotic lower bound for $f(n)$.

- $\Theta$ is stronger than $\Omega$:
  $$f(n) = \Theta(g(n)) \Rightarrow f(n) = \Omega(g(n)),$$
  or $\Theta(g(n)) \subseteq \Omega(g(n))$

- We write $f(n) = \Omega(g(n))$ for $f(n) \in \Omega(g(n))$
### Examples for $\Omega$

- $a \ n^2 + b \ n + c = \Omega(n^2)$ provided $a > 0$
  
  since it is also $\Theta(n^2)$.

- $a \ n^2 + b \ n + c = \Omega(n)$ provided $a > 0$

- $\Omega$ can be used for a lower bound of the running time for best-case input (and hence for any input). For example, the best-case running time of Insertion-Sort is $\Omega(n)$.

### Properties of $\Theta$, $O$, $\Omega$

- $f(n)$ is a tight bound if it is an upper and lower bound:
  
  $f(n) = \Theta(n) \iff f(n) = O(n) \land f(n) = \Omega(n)$

- $f(n) = O(g(n)) \land g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$
  
  $f(n) = \Omega(g(n)) \land g(n) = \Omega(h(n)) \Rightarrow f(n) = \Omega(h(n))$  
  (Transitivity)

- $f(n) = O(f(n))$
  
  $f(n) = \Omega(f(n))$  
  (Reflexivity)

- $f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$  
  (Transpose Symmetry)
Example for O, Ω, Θ

- $3n^2 - 100n + 6 = O(n^2)$ because $3n^2 > 3n^2 - 100n + 6$
- $3n^2 - 100n + 6 = O(n^3)$ because $.00001n^3 > 3n^2 - 100n + 6$
- $3n^2 - 100n + 6 \neq O(n)$ because $cn < 3n^2$ when $n > c$
- $3n^2 - 100n + 6 = \Omega(n^2)$ because $2.99n^2 < 3n^2 - 100n + 6$
- $3n^2 - 100n + 6 \neq \Omega(n^3)$ because $3n^2 - 100n + 6 < n^3$
- $3n^2 - 100n + 6 = \Omega(n)$ because $10^{10}n < 3n^2 - 100 + 6$
- $3n^2 - 100n + 6 = \Theta(n^2)$ because both $O$ and $\Omega$
- $3n^2 - 100n + 6 \neq \Theta(n^3)$ because not $\Omega$
- $3n^2 - 100n + 6 \neq \Theta(n)$ because not $O$

Asymptotic Notation in Equations

- $f(n) = \Theta(n)$ simply means $f(n) \in \Theta(n)$

- More generally, $\Theta(n)$ stands for an anonymous function which is an element of $\Theta(n)$, e.g.
  - $3n^2 + 3n + 1 = 2n^2 + \Theta(n)$
  - means
  - $3n^2 + 3n + 1 = 2n^2 + f(n) \land f(n) \in \Theta(n)$ for some $f$

- In recurrences:
  - $T(n) = 2T(n/2) + \Theta(n)$

- In calculations:
  - $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$
  - $= \Theta(n^2)$
\section*{\textit{o-Notation}}

- The upper bound provided by \( O \) may or may not be tight. We use \( o \) for an upper bound which is not tight.

- Definition: Let \( g(n) \) be an asymptotically non-negative function on the natural numbers.
\[
o(g(n)) = \{ f(n) \mid \forall c > 0 \cdot \exists n_0 \in \text{Nat} \cdot \\
\quad \forall n \geq n_0 \cdot 0 \leq f(n) \leq c g(n) \}
\]
The idea of the definition is that \( f(n) \) becomes insignificant relative to \( g(n) \) as \( n \) approaches infinity.

- For example:
  - \( 2n = o(n^2) \)
  - \( 2n^2 \neq o(n^2) \)
  - \( 2n^3 \neq o(n^2) \)

\section*{\textit{\( \omega \)-Notation}}

- The lower bound provided by \( \Omega \) may or may not be tight. We use \( \omega \) for a lower bound which is not tight.

- Definition: Let \( g(n) \) be an asymptotically non-negative function on the natural numbers.
\[
\omega(g(n)) = \{ f(n) \mid \forall c > 0 \cdot \exists n_0 \in \text{Nat} \cdot \\
\quad \forall n \geq n_0 \cdot 0 \leq c g(n) \leq f(n) \}
\]
The idea of the definition is that \( g(n) \) becomes insignificant relative to \( f(n) \) as \( n \) approaches infinity.

- For example:
  - \( n^2 / 2 = \omega(n) \)
  - \( n^2 / 2 \neq \omega(n^3) \)
  - \( n^2 / 2 \neq \omega(n^3) \)
Example: Factorial

- \( n! \) is defined by:
  \[
  0! = 1 \\
  n! = (n-1)! \text{ for } n > 0
  \]

- From Stirling's approximation
  \[
  n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)
  \]
  one can derive:
  - \( n! = o(n^n) \)
  - \( n! = \omega(2^n) \)